Solar System

Dienesch Florian | Rathbauer Alexander

Inhaltsverzeichnis

[Aufgabenstellung 2](#_Toc412449345)

[Zeitaufzeichnung 3](#_Toc412449346)

[Arbeitsschritte 3](#_Toc412449347)

[Skizze Layout 3](#_Toc412449348)

[UML – Klassendiagramm 3](#_Toc412449349)

[Implementierung 3](#_Toc412449350)

[Probleme 3](#_Toc412449351)

[Testfälle 3](#_Toc412449352)

[Entwicklungsumgebung 3](#_Toc412449353)

[Tools 3](#_Toc412449354)

[Quellen 3](#_Toc412449355)

# Aufgabenstellung

Wir wollen nun unser Wissen aus Medientechnik und SEW nützen um eine etwas kreativere Applikation zu erstellen.

Eine wichtige Library zur Erstellung von Games mit 3D-Grafik ist Pygame. Die 3D-Unterstützung wird mittels PyOpenGL erreicht.

Die Kombination ermöglicht eine einfache und schnelle Entwicklung.

Während pygame sich um Fensteraufbau, Kollisionen und Events kümmert, sind grafische Objekte mittel OpenGL möglich.

Die Aufgabenstellung:

Erstellen Sie eine einfache Animation unseres Sonnensystems:

In einem Team (2) sind folgende Anforderungen zu erfüllen.

* Ein zentraler Stern
* Zumindest 2 Planeten, die sich um die eigene Achse und in elliptischen Bahnen um den Zentralstern drehen
* Ein Planet hat zumindest einen Mond, der sich zusätzlich um seinen Planeten bewegt
* Kreativität ist gefragt: Weitere Planeten, Asteroiden, Galaxien,...
* Zumindest ein Planet wird mit einer Textur belegt (Erde, Mars,... sind im Netz verfügbar)

Events:

* Mittels Maus kann die Kameraposition angepasst werden: Zumindest eine Überkopf-Sicht und parallel der Planentenbahnen
* Da es sich um eine Animation handelt, kann diese auch gestoppt werden. Mittels Tasten kann die Geschwindigkeit gedrosselt und beschleunigt werden.
* Mittels Mausklick kann eine Punktlichtquelle und die Textierung ein- und ausgeschaltet werden.
* Schatten: Auch Monde und Planeten werfen Schatten.

Hinweise:

* Ein Objekt kann einfach mittels glutSolidSphere() erstellt werden.
* Die Planten werden mittels Modelkommandos bewegt: glRotate(), glTranslate()
* Die Kameraposition wird mittels gluLookAt() gesetzt
* Bedenken Sie bei der Perspektive, dass entfernte Objekte kleiner - nahe entsprechende größer darzustellen sind.  
  Wichtig ist dabei auch eine möglichst glaubhafte Darstellung. gluPerspective(), glFrustum()
* Für das Einbetten einer Textur wird die Library Pillow benötigt! Die Community unterstützt Sie bei der Verwendung.

 Tutorials:

* Pygame: <https://www.youtube.com/watch?v=K5F-aGDIYaM>

Viel Erfolg!

# Zeitaufzeichnung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User Story | Verantwortlicher | Zeit [std] | | Status |
| Gesch. | Wirkl |
| Einarbeitung | D,R | 4 |  |  |
| UML | R | 3 |  |  |
| Galaxy anzeigen | D | 1 |  |  |
| Modelle erstellen | R | 0.5 |  |  |
| Modelle um sich selbst drehen | R | 4 |  |  |
| Modelle um andere Modelle drehen | R | 4 |  |  |
| Modelle in ellipsenbahn drehen lassen | D | 4 |  |  |
| Animation | D | 1 |  |  |
| Geschwindigkeit änderbar | R | 1 |  |  |
| Licht | D | 6 |  |  |
| Texturen | R | 6 |  |  |
| Steuerbare Kamera | D | 3 |  |  |
| Hintergrund | R | 1.5 |  |  |
| Tests | D | 1.5 |  |  |
| Zusammenfassung |  | 40.5 |  |  |

# Arbeitsschritte

## Skizze Layout

## UML – Klassendiagramm

## Implementierung

# Probleme

# Testfälle

# Entwicklungsumgebung

## Tools

# Quellen